

10/26/99
JCS72 U.S. PTO

Express Mail Label: EL278372858US

Please type a plus sign (+) inside this box ☐

Approved for use through 02/30/2000. CMB 0051-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL	
(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))	
Attorney Docket No.	RCA 89,086
First Inventor or Application Identifier	Kevin Lloyd Grimes et al.
Title	AN ADAPTIVE TRANSPORT PROTOCOL DECODER
Express Mail Label No.	EL278372858US

APPLICATION ELEMENTS		ADDRESS TO: Assistant Commissioner for Patents Box Patent Application Washington, DC 20231	
See MPEP chapter 800 concerning utility patent application contents.			
1. <input checked="" type="checkbox"/> * Fee Transmittal Form (e.g., PTO/SB/17) (Submit an original and a duplicate for fee processing)	5. <input type="checkbox"/> Microfiche Computer Program (Appendix)		
2. <input checked="" type="checkbox"/> Specification [Total Pages 21] (preferred arrangement set forth below)	6. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)		
- Descriptive title of the Invention	a. <input type="checkbox"/> Computer Readable Copy		
- Cross References to Related Applications	b. <input type="checkbox"/> Paper Copy (identical to computer copy)		
- Statement Regarding Fed sponsored R & D	c. <input type="checkbox"/> Statement verifying identity of above copies		
- Reference to Microfiche Appendix			
- Background of the Invention			
- Brief Summary of the Invention			
- Brief Description of the Drawings (if filed)			
- Detailed Description			
- Claim(s)			
- Abstract of the Disclosure			
3. <input checked="" type="checkbox"/> Drawings (35 U.S.C. 113) [Total Sheets 5]			
4. Oath or Declaration [Total Pages 2]			
a. <input checked="" type="checkbox"/> Newly executed (original or copy)			
b. <input type="checkbox"/> Copy from a prior application (37 C.F.R. § 1.63(d)) (for continuation/divisional with Box 16 completed)			
i. <input type="checkbox"/> DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).			

NOTE: FOR ITEMS 1 & 13, IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.37), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).

ACCOMPANYING APPLICATION PARTS	
7. <input checked="" type="checkbox"/> Assignment Papers (cover sheet & document(s))	
8. <input type="checkbox"/> 37 C.F.R. § 73(b) Statement (when there is an assignee)	<input checked="" type="checkbox"/> Power of Attorney
9. <input type="checkbox"/> English Translation Document (if applicable)	
10. <input checked="" type="checkbox"/> Information Disclosure Statement (IDS)/PTO-1449	<input type="checkbox"/> Copies of IDS Citations
11. <input type="checkbox"/> Preliminary Amendment	
12. <input checked="" type="checkbox"/> Return Receipt Postcard (MPEP 503) (Should be specifically itemized)	
13. <input type="checkbox"/> Small Entity Statement(s)	<input type="checkbox"/> Statement filed in prior application, Status still proper and desired
14. <input type="checkbox"/> Certified Copy of Priority Document(s) (if foreign priority is claimed)	
15. <input type="checkbox"/> Other:	

16. IF A CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: _____

Prior application information: Examiner: _____ Group / Art Unit: _____

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

17. CORRESPONDENCE ADDRESS

☐ Customer Number or Bar Code Label (Insert Customer No. or Attach bar code label here) or ☐ Correspondence address below

Name	Joseph S. Tripoli		
	Thomson Multimedia Licensing Inc.		
Address	Patent Operation		
	Two Independence Way, P. O. Box 5312		
City	Princeton	State	NJ
Country	USA	Telephone	609/734-9701
		Zip Code	08543-5312
		Fax	609/734-9700

Name (Print/Type)	Ronald H. Kurdyla	Registration No. (Attorney/Agent)	26,932
Signature	<i>R. Kurdyla</i>	Date	10/26/99

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

AN ADAPTIVE TRANSPORT PROTOCOL

This is a non-provisional application originating from a provisional application, serial number 60/142,849, filed July 8, 1999.

The present invention relates to a transport packet decoder which adaptively decodes packets having differing transport protocols.

Currently, digital signals carrying programming, such as video/audio/data programming, are supplied to consumers from different providers in respectively different formats, called protocols. For example, direct satellite system (DSS) signals are formatted in a proprietary format owned by DirecTV, and all signals carrying programming supplied via DSS satellites are formatted using that protocol. Similarly, local terrestrial high definition television (HDTV) signals are formatted according to a standard initially proposed by the Advanced Television Standards Committee (ATSC) and approved by the US Federal Communications Commission (FCC), and all signals carrying HDTV programming are formatted using that standard protocol.

Consumers will want to receive digital signals of these and any other protocols through which digital signals carrying programming is carried. Currently, this requires separate decoders, each embodied in a separate enclosure, such as a set top box, for each desired protocol. This is expensive for the consumer, requires a large amount of space for the various set top boxes, and is inconvenient to use. For example, each set top box may have its own remote control which may be incompatible with those for the other set top boxes. Thus, a transport protocol decoder which can decode different protocols is desirable.

The present inventors have realized that the protocols, such as those described above, each specify a stream of transport packets of different formats, e.g. having different sizes, different framing, different bit arrangement, etc.

Further, each of the transport packets carry payloads containing digital data representing signals of different types, such as encoded video or audio, or data. However, the inventors have also realized that the signals represented by the data in the respective payloads for each of the signal types, are the same for all of the protocols. That is, for both of the DSS and ATSC protocols, the encoded video and audio signals are MPEG encoded signals. Consequently, once the payloads have been extracted from the received transport packets, the remainder of the signal processing for such signals is the same regardless of the protocol of the received packets.

In accordance with principles of the present invention, an adaptive transport protocol decoder includes a source of a stream of packets, each including a payload, and having a first transport protocol, and a source of a stream of packets, each including a payload, and having a second transport protocol. A protocol decoder, coupled to the first and second packet stream sources, extracts the respective payloads from the packets from a selected one of the first and second packet stream sources.

In the drawing:

Fig. 1 is a block diagram of a system including an adaptive transport decoder according to the present invention;

Fig. 2 is a diagram illustrating the format of two packets having different protocols useful in understanding the operation of the adaptive transport decoder of Fig. 1;

Fig. 3 is a more detailed block diagram of an adaptive transport protocol decoder which may be used in the system illustrated in Fig. 1;

Fig. 4 is a memory layout diagram illustrating the various software components which may be used in an embodiment of an adaptive transport protocol decoder illustrated in Fig. 3;

Fig. 5 is a memory layout diagram illustrating the various software components illustrated in Fig. 4 arranged for processing DSS transport protocol packets;

Fig. 6 is a memory layout diagram illustrating the various software components illustrated in Fig. 4 arranged for processing ATSC transport protocol packets;

Fig. 7 is a flow chart illustrating the process of switching modes between decoding DSS transport packets to decoding ATSC transport packets.

Fig. 1 is a block diagram of a system including an adaptive transport decoder according to the present invention. In Fig. 1, respective output terminals of a plurality 10 of sources of streams of packets are coupled to corresponding input terminals of a selector 20. An output terminal of the selector 20 is coupled to an input terminal of an adaptive transport protocol decoder 30. An output terminal of the adaptive transport protocol decoder 30 is coupled to an input terminal of a payload processor 40. An output terminal of the payload processor 40 is coupled to an input terminal of a user interface 50.

In operation, packet source 12 produces a stream of packets each including a payload (described in more detail below) and having a first transport protocol, and packet source 14 produces packets each including a payload and having a second transport protocol. The plurality 10 of packet stream sources may include additional packet stream sources, e.g. packet stream source 1N, producing respective streams of packets each including a payload and having respectively different transport protocols.

For example, the packet source 12 may be the front end of a DSS satellite signal receiver, including a receiving antenna, RF and IF receiver circuitry, tuner, forward error correction (FEC) circuitry, packet error detection and/or correction circuitry (Viterbi or Reed-Solomon decoding, etc.), decryptor and so forth, none of which is germane to the invention, and thus are neither shown nor described in detail. The output from the packet stream source 12 is a stream of packets representing the DSS signal each having the DSS transport packet protocol (format). Similarly, the packet source 14 may be the front end of a terrestrial HDTV signal receiver, including an antenna or cable connection, RF and IF receiver circuitry, error detection and/or correction circuitry and so forth (not shown). The output from the packet stream source 14 is a stream of packets representing the HDTV signal each having the ATSC transport packet protocol.

The selector 20 selects one of the plurality 10 of packet stream sources in response to a control signal from a controller (not shown) and supplies the packet stream from the selected packet stream source to the adaptive transport protocol decoder 30, in a known manner. The adaptive transport protocol decoder 30 analyzes each packet as it is received, and performs appropriate processing of that packet, depending on the transport protocol of the packet, to extract the payload data from the packet in a manner to be described in more detail below. The extracted payload data is supplied to the payload processor 40.

The payload processor 40 processes the payload data to generate a signal corresponding to that data, and supplies that signal to the user interface 50. For example, if the extracted payload data from the adaptive transport protocol adapter 30 is MPEG2 encoded digital video data, then the payload processor 40 includes a known MPEG decoder which decodes the MPEG2 video data according to the MPEG standard to generate a standard video signal (e.g. an NTSC video signal in the US). The video signal is then supplied to the user interface 50, which in this case would be a video monitor, which displays the image represented by the video signal.

Fig. 2 is a diagram illustrating the format of two packets having different protocols. The packets illustrated in Fig. 2 are illustrative only, and not meant to represent a packet of any known protocol. In Fig. 2, a packet having a first protocol is illustrated as a rectangle labeled A), and a packet having a second protocol is illustrated as a rectangle labeled B). The length of the rectangles is an indication of the relative sizes (number of bits) of the packets, and similarly the lengths of the subsections within the packet is an indication of the relative sizes (number of bits) of that subsection. As illustrated in Fig. 2, each of packets A) and B) contains a fixed predetermined number of bits, with the number of bits in packet A) being different from (e.g. less than) the number of bits in packet B). Both packets A) and B) consist of a header, containing control information (described in more detail below), and a payload containing the data carried by the packet. Also as illustrated in Fig. 2, the payloads of packets A) and B) contain a fixed predetermined number of bits, with the number of bits in the payload of packet A) being different from (e.g. less than) the number of bits in the payload of packet B).

The control information in the header is used (among other things) to identify the content of and route the data contained in the packet payload. In Fig. 2, the format of the header information is displayed in expanded form for the packets A) and B). For example, each payload carries data representing a signal of a particular type, e.g. video, audio, data, which must be processed differently from the other types. Control information (TYPE) indicating the type of data in the packet payload is included in the packet headers of the packets A) and B). Also in the packet stream, are data streams carrying data representing a single signal and formed by concatenating the payloads of those packets carrying that data stream. For example, in a DSS channel carrying a television program, there may be several audio signals representing different languages, or several data channels representing closed captioning in different languages, or several video channels representing images from different camera angles or parental approval levels. Each such data stream is identified by a unique packet identifier (PID),

which is included in the packet headers for packets A) and B). Further, in order to facilitate pay-per-view channels, the payload data in the packet may or may not be encrypted. Each of the headers of packets A) and B) contain control information (E) to indicate whether the payload data in that packet requires
5 decrypting.

Fig. 3 is a more detailed block diagram of an adaptive transport protocol decoder 30 which may be used in the system illustrated in Fig. 1. The adaptive transport protocol decoder 30 of Fig. 3, includes an input terminal coupled to the
10 output terminal of the selector 20 (of Fig. 1). The input terminal of the adaptive transport protocol decoder 30 is coupled to the series connection of a service detector 31, a decryptor 33, transport decoder 35, buffer memory 37 and application control circuit 39. A controller 32, which may include a microprocessor (not shown), has a control signal output terminal coupled to
15 respective control input terminals of the service detector 31, the decryptor 33, the transport decoder 35, the buffer memory 37 and the application control circuit 39. A memory bus is coupled between the controller 32 and an internal memory 34 (which may include both read/write memory (RAM) and read-only memory (ROM)) and an external memory 34'. The service detector 31,
20 decryptor 33, transport decoder 35, buffer memory 37, application control circuit 39, controller 32 and internal memory 34 may be embodied within an integrated circuit (IC) chip, while the external memory 34' resides outside the IC chip).

25 In operation, the service detector 31 receives the selected packet stream from the selector 20 (as described above). The service detector 31 detects the start and end of every packet, in a known manner. For an ATSC encoded MPEG2 packet stream, for example, packets are delimited by synchronization
30 bytes having a predetermined value, e.g. 47h, at predetermined locations in the packet stream. Referring to Fig. 2, such a synchronization byte (SB) is illustrated in phantom in packet A) because the synchronization byte SB is not part of the packet itself. The service detector 31 may detect the beginnings and ends of

packets using the synchronization bytes. However in DSS packet streams, no synchronization bytes are present. For DSS packet streams, the start and end of packets are detected by means of a synchronization detector in the selected packet stream source 10 which generates a signal to indicate the start of each packet, all in a known manner. For such packet streams, the service detector 31 may use this signal to detect the beginnings and endings of the packets. In the illustrated embodiment a synchronization byte (47h) is inserted between DSS packets by the service detector 31. Because all packets are separated by synchronization bytes, downstream processing will be more uniform for the different protocols.

When each packet has been separated, the service detector 31 then determines if the received packet belongs to a data stream currently being received. For instance, though there may be several audio data streams in the selected packet stream, usually only one or two will be received and further processed. Specifically, in the illustrated embodiment, up to eight data streams may be received. However, one skilled in the art will understand that there is no inherent limit to the number of data streams which may be received simultaneously. As described above, the PID in the header of each packet identifies the data stream to which the packet belongs. The receiver system maintains a set of registers which contain the PIDs of the data streams currently being received and processed. For each received packet, the service detector 31 extracts the PID. As described above, the PID may be placed in a different place, have a different length, and be encoded in a different manner in different protocols. In a manner to be described in more detail below, the service detector 31, in response to a control signal from the controller 32, extracts the PID from the header location appropriate for the protocol of the packet. The extracted PID is stored in a register within the adaptive transport protocol decoder 30. This is illustrated in Fig. 2 by curved lines from the location of the PIDs in packets A) and B) to the PID register. The remainder of the processing to determine whether the data stream carried by this packet is to be further processed operates on the data in the PID register. For example, the PID in the PID register

may be compared to the PIDs in the set of registers in the receiver system. If there is a match, the packet is one that is being further processed, and it is then processed by the decryptor 33, otherwise the packet is ignored and the next packet received is processed as described above.

5

The decryptor 33 extracts the header information indicating whether the payload of this packet is encrypted. As described above, the encryption information E may be placed in a different place, have a different length, and be encoded in a different manner in different protocols. In a manner to be described in more detail below, the decryptor 33, in response to a control signal from the controller 32, extracts the encryption information E from the header location appropriate for the protocol of the packet. Similar to the service detector 31, the extracted encryption information E is stored in a register within the adaptive transport protocol decoder 30, as illustrated by the curved lines from the location of the encryption information E in packets A) and B) to the E register. The remainder of the processing to determine whether the packet needs to be decrypted is performed on the data in the E register. If the encryption information E indicates that the packet requires decrypting, the decryptor 33 decrypts the packet to generate a plain-text packet which may be further processed. Otherwise, the decryptor 33 passes the packet without change to the transport decoder 35.

25

The transport decoder 35 extracts the payload from the received packet from its location in the received packet. As described above, the payload may be placed in a different place and have a different length in different protocols. The transport decoder 35 places the extracted payload in a buffer memory 37. The application control circuit 39, then, retrieves the buffered payload from the buffer memory and routes them to appropriate payload processing circuitry. In a manner to be described in more detail below, the application control circuit 39, in response to a control signal from the controller 32, extracts the data stream type information TYPE from the header location appropriate for the protocol of the packet. The extracted data stream type information TYPE is stored in a register

30

within the adaptive transport protocol decoder 30, as illustrated by the curved lines from the locations of the data stream type information TYPE in packets A) and B) to the TYPE register. The remainder of the processing to determine where the payload of this packet should be routed is performed on the data in the TYPE register. Specifically, the data in the TYPE register is read and evaluated, and the application control circuit 39 routes the payload to an appropriate processor in the payload processor 40 (of Fig. 3). If, for example, the payload carries MPEG encoded video data, the payload is supplied to an MPEG decoder. If the payload carries closed captioning information, the payload is supplied to an on screen display generator, and so forth.

Fig. 3 illustrates only an exemplary and minimal set of circuitry for processing received transport packets. However, one skilled in the art will understand that other processing may be required to properly decode transport packets. For example, packet streams may include transport packets which carry auxiliary data. Auxiliary data is not payload data part of a data stream, but instead carries status and control information for use by the receiver system itself. In addition, some protocols may allow a combination of payload and auxiliary data in a single transport packet. Further, different protocols may include other information in the header and carry data in respectively different manners, requiring correspondingly different processing. One skilled in the art of decoding transport packets in those protocols will understand what other processing is required by those protocols, what processing elements are necessary to provide that processing, and how to design and implement those processing elements.

Fig. 4 is a layout diagram for the memory 34/34' in the adaptive transport protocol decoder 30 illustrated in Fig. 3, illustrating the various software components (storage and code). The arrangement in Fig. 4 is schematic only and is not meant to indicate relative or absolute location or size of the illustrated components, nor their placement in either of the memories 34 or 34'. In Fig. 4, the operation of the adaptive transport protocol adapter 30 is invoked by an

external, hardware-generated, interrupt request (INT REQ) signal generated by the selected packet stream source 10 when a packet is available. The interrupt request signal (INT REQ) is supplied to the adaptive transport protocol decoder 30 via the selector 20. This interrupt request signal (INT REQ) is processed by the interrupt mechanism of a microprocessor (not shown) embodied in the controller 32. The interrupt request (INT REQ) signal causes the microprocessor to interrupt its processing, and to immediately begin processing of a packet handler 104 at a memory location pointed to by an interrupt pointer PNTR(O) associated with the external hardware interrupt request signal (INT REQ) in an interrupt vector 102 stored at a predetermined fixed location in the memory, all in a known manner. The interrupt pointer PNTR(O) contains a memory location of the packet (interrupt) handler 104 which is designated to process this interrupt. The interrupt handler 104 performs the processing necessary to determine if the available packet is to be further processed, to decrypt the packet if necessary, extract the payload, and store the extracted payload in a buffer until it can be further processed, (and any other required processing) as described above. It would also be possible for processing of a newly available packet to be initiated by status polling by the microprocessor, or any other of known methods.

When the microprocessor (not shown) is interrupted in this manner, the current state of the microprocessor program execution (program counter, registers, flags, etc.) are pushed onto a stack 110, and the interrupt routine 104 pointed to by the interrupt pointer PNTR(O) in the interrupt vector 102 is executed starting at the address in the pointer PNTR(O). Because this is a rapid way to switch program environments, the interrupt process described above may also be invoked via a software instruction, termed a software interrupt, in time critical applications. For example, a routine DVR(1) may be invoked by a software interrupt calling the routine pointed to by the interrupt pointer PNTR(1); a routine DVR(2) may be invoked by a software interrupt calling the routine pointed to by the interrupt pointer PNTR(2), and so forth. In Fig. 4, the interrupt handler 104 invokes the various driver routines 106 via software interrupts

directed to the appropriate interrupt pointers PNTR(x) in the interrupt vector 102. This is indicated in Fig. 4 via a thick arrow from the interrupt handler 104 to the interrupt vector 102. It is further possible (though not illustrated in Fig. 4 to simplify the drawing) for driver routines DVR(x) to call other driver routines

5 DVR(x) through software interrupts specifying the interrupt pointer PNTR(x) in the interrupt vector 102 of the desired driver routine DVR(x).

In Fig. 4, the buffer memory 37 includes sufficient memory (BUFFER(1) through BUFFER(4)) for four extracted payloads. As a payload is extracted from

10 a packet, it is stored in the next available buffer and the payload processor 40 appropriate for that payload is notified of the location in the buffer memory 37 of the payload. In response, the payload processor 40 retrieves the payload from its buffer, and processes it in an appropriate manner.

In the illustrated embodiment, the adaptive transport protocol decoder 30 and its memory 34/34' are able to be configured to process either DSS transport protocol packets, or ATSC transport protocol packets. The design and implementation of an interrupt handler 104 and driver routines 106 necessary to process DSS protocol packets; the instructions making up the interrupt handler

15 104 and driver routines 106; and the addresses of the interrupt handler 104 and the driver routines 106 for the interrupt vector 102, are all known and will not be described in detail. In addition, the size of the buffers in the buffer memory 37 is also known: the payload of DSS transport packets are 127 bytes in length. Similarly, the particular routines necessary to process ATSC protocol packets,

25 the instructions making up the interrupt handler 104 and the driver routines 106, the addresses of the interrupt handler 104 and the driver routines 106 for the interrupt vector 102 are all known and will not be described in detail. In addition, the size of the buffer memory 37 is also known: the payload of ATSC transport packets are 184 bytes in length.

Fig. 5 is a memory layout diagram illustrating the various software components illustrated in Fig. 4 arranged for processing DSS transport protocol

packets. In Fig. 5, those elements which are the same as those illustrated in Fig. 4 are designated by the same reference number and are not described in detail. As in Fig. 4, the arrangement in Fig. 5 is schematic only and is not meant to indicate relative or absolute location or size of the illustrated components.

There is a limited amount of memory 34 internal to the adaptive transport protocol decoder 30 IC chip. Thus, some portion of the software components must be located in the external memory 34'. In Fig. 5, the DSS driver routines DVR(1) through DVR(N) are located in the control read-only memory (ROM) 342 of the internal memory 34. The DSS vector table 102, stack 110 and payload buffers 37 are located completely in the internal RAM 344 of the internal memory 34. Due to memory size constraints, however, only a portion of the DSS interrupt handler 104(1) is located in the internal RAM 344. The remainder of the DSS interrupt handler 104(2) is located in the external dynamic RAM (DRAM) 34'. As described above, the design, implementation and operation of the DSS software components to decode DSS protocol transport packets is known, and will not be described here.

It may also be seen in Fig. 5 that the ATSC interrupt vector 102', software interrupt drivers 106' and interrupt handler 104' for controlling the operation of the adaptive transport protocol decoder 30 when decoding ATSC transport packets are stored in the external DRAM 34', but are not utilized.

In operation, when the DSS interrupt handler 104 is invoked by an external interrupt signal (not shown). The DSS interrupt handler 104 invokes appropriate DSS driver routines 106 to process the received transport packet, extracting data from the header, decrypting the payload if necessary, storing the payload in one of the buffers 37, and notifying the appropriate payload processor 40 under control of the controller 32, as described above, all in a known manner.

Fig. 6 is a memory layout diagram illustrating the various software components illustrated in Fig. 4 arranged for processing ATSC transport protocol

packets. In Fig. 6, those elements which are the same as those illustrated in Figs. 4 and 5 are designated by the same reference number and are not described in detail. As in Figs. 4 and 5, the arrangement in Fig. 6 is schematic only and is not meant to indicate relative or absolute location or size of the illustrated components.

As in Fig. 5, some portion of the software components must be located in the external memory 34'. In Fig. 6, the interrupt vector 102', and payload buffers 37 are located completely in the internal RAM 344 of the internal memory 34. Due to memory size constraints, however, only a portion of the stack 110(1) is located in the internal RAM 344. The remainder of the stack 110(2) is located in the external DRAM 34'. The ATSC interrupt handler 104' and ATSC driver routines 106' are completely located in the external DRAM 34'. As described above, the design, implementation and operation of the ATSC software components to decode ATSC protocol transport packets is known, and will not be described here.

It may also be seen in Fig. 6 that the DSS interrupt drivers 106 remain located in the control ROM 342 in the internal memory 34. The DSS interrupt vector 102 and interrupt handler 104 are stored in the external DRAM 34'. None of these are utilized.

In operation, when the ATSC interrupt handler 104 is invoked by an external interrupt signal (not shown). The ATSC interrupt handler 104 invokes the appropriate interrupt drivers to process the received transport packets, extracting data from the header, decrypting the payload if necessary, storing the payload in one of the buffers 37, and notifying the appropriate payload processor 40 under control of the controller 32, as described above, all in a known manner.

It is also possible for an interrupt driver in the DSS interrupt drivers 106 in the control ROM 342 of the internal memory 34 to perform a function which is required when decoding ATSC transport packets. In Fig. 6, the ATSC interrupt

pointer PNTR(2) points to an interrupt driver DVR(2) in the DSS interrupt driver set 106. The function of this driver, consequently, need not be recreated in the ATSC interrupt driver set 106'. For example, as described above, information is extracted from the header of a packet and placed in registers for further processing. An interrupt driver DVR(n) may process a payload type by accessing that register regardless of where the information came from in the header. Such an interrupt driver DVR(n) may then be shared among all the protocols which the adaptive transport protocol decoder 30 is designed to process. More specifically, as described above, the PID is extracted from the header of the packet and placed in the PID register. The contents of that register is then compared to a table of PIDs identifying data streams currently being received. Once the PID has been extracted from the header of the packet and placed in the register, the interrupt driver which compares the PID in the register to the table of PIDs can be shared among all the transport packet protocol control programs.

Fig. 7 is a flow chart illustrating the process of switching modes between decoding DSS transport packets to decoding ATSC transport packets. A switch between decoding DSS transport packets and ATSC transport packets may be initiated by any of a variety of known manners. For example, referring to Fig. 1, under user control, the user selects a packet stream whose programming it is desired to watch. If the packet stream source 12 produces packets in DSS protocol, then when packet stream source 12 is coupled to the adaptive transport protocol decoder 30 by the selector 20, the adaptive transport packet decoder is simultaneously configured to decode DSS transport packets. If the packet stream source 14 produces packet in the ATSC protocol, then when packet stream source 14 is coupled to the adaptive transport protocol decoder 30 by the selector 20, the adaptive transport protocol decoder 30 is simultaneously configured to decode ATSC transport packets. Alternatively, several on-line program directories exist, which can carry the protocol of the listed programs. When a program is selected, the directory is consulted and the adaptive transport protocol decoder 30 is configured accordingly. Other

methods are also available for configuring the adaptive transport protocol decoder 30.

In Fig. 7, a switch of protocols is initiated in step 202. In step 204,
5 memory is reallocated for the payload buffers 37. For example, DSS payload buffers contain 127 bytes and ATSC payload buffers contain 184 bytes. In step 206, the software (S/W) pointers are adjusted. In the illustrated embodiment, when the adaptive transport protocol decoder 30 is configured to decode DSS packets, the DSS interrupt vector 102 (stored in the external DRAM 34' while
10 the adaptive transport protocol decoder 30 is configured to process ATSC transport packets) is moved to the internal RAM 344 of the internal memory 34, while the ATSC interrupt vector 102' is simultaneously moved from the internal RAM 344 of the internal memory 34 to the external DRAM 34'. Then memory is allocated for the stack 110. Finally, the portion of the DSS interrupt handler
15 104(1) which is to be located in the internal RAM 344 of the internal memory 34 is moved to its allocated location.

Conversely, when the adaptive transport protocol decoder 30 is configured to decode ATSC packets, the ATSC interrupt vector 102' (stored in
20 the external DRAM 34' while the adaptive transport protocol decoder 30 is configured to process DSS transport packets) is moved to the internal RAM 344 of the internal memory 34, while the DSS interrupt vector 102 is simultaneously moved from the internal RAM 344 of the internal memory 34 to the external DRAM 34'. Then memory is allocated for the portion of the stack 110(1) which
25 is to be located in the internal RAM 344 of the internal memory 34.

In step 208, the registers (of Fig. 1) which hold various information extracted from the headers of the received packets are cleared. In step 210, the switch process is completed, and the adaptive transport protocol decoder 30 is
30 ready to decode packets in the newly selected protocol.

The adaptive transport protocol decoder 30 has been described in terms of two transport protocols: DSS and ATSC. One skilled in the art will understand that additional protocols may be implemented by appropriate design and coding of an interrupt handler 104 and interrupt drivers 106 to process headers and
5 extract payloads from packets of those protocols, and then reconfiguring the adaptive transport protocol decoder 30 to use the appropriate interrupt handler 104 and interrupt drivers 106 via coding of the interrupt vector 102 for that protocol. Use of software interrupts for performing the required processing produces fast context switching, and fast and easy reconfiguration of the
10 adaptive transport protocol adapter 30.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208

What is claimed is:

1. An adaptive transport protocol decoder, comprising:

5 a source of a stream of packets, each including a payload, and having a first transport protocol;

a source of a stream of packets, each including a payload, and having a second transport protocol;

10 a protocol decoder, coupled to the first and second packet stream sources, for extracting the respective payloads from the packets from a selected one of the first and second packet stream sources.

15 2. The adaptive transport protocol decoder of claim 1 further comprising a selector, having respective input terminals coupled to the first and second packet stream sources, and an output terminal coupled to the protocol decoder, and responsive to a select signal for coupling one of the first and second packet stream sources to the protocol decoder.

20 3. The adaptive transport protocol decoder of claim 1 wherein the protocol decoder comprises a processor, responsive to a first control program for processing the packets from the first packet stream source to extract the respective payloads, a second control program for processing the packets from
25 the second packet stream source to extract the respective payloads, and a third control program for switching between the first control program and the second control program.

30 4. The adaptive transport protocol decoder of claim 3 wherein:

the processor includes a memory for storing the first, second and third programs; and

both of the first and second control programs comprise:

a packet handler, executed in response to each received packet;

5

a plurality of interrupt drivers, stored in the memory at respective locations, called by software interrupt;

an interrupt vector, stored at a fixed, predetermined location in memory,
10 including a plurality of entries, each containing a pointer to a respective location of an interrupt driver.

5. The adaptive transport protocol decoder of claim 4 wherein the third control program switches between the first and second control programs by
15 moving the interrupt vector of one of the first and second control programs to the fixed predetermined location in the memory, and simultaneously moving the interrupt vector of the other one of the first and second control programs to another location in the memory.

20 6. The adaptive transport protocol decoder of claim 4 wherein both the first and second control programs further comprise a buffer for storing the respective extracted payloads at a location in the memory.

25 7. The adaptive transport protocol decoder of claim 6 wherein the third control program switches between the first and second control programs by:

moving the interrupt vector of one of the first and second control programs to the fixed predetermined location in the memory, and simultaneously moving the interrupt vector of the other one of the first and second control programs to another location in the memory; and
30

reallocating the buffer to a location in the memory.

8. The adaptive transport protocol decoder of claim 4 wherein:

the packet handler is an interrupt handler stored in the memory at a
5 location; and

one of the entries in the interrupt vector points to the location of the
packet handler.

10 9. The adaptive transport protocol decoder of claim 8 wherein:

each of the first and second packet stream sources generates an interrupt
request signal when a packet is available;

15 the entry in the interrupt vector pointing to the location of the packet
handler is responsive to the interrupt signal from the selected packet stream
source.

20 10. The adaptive transport protocol decoder of claim 3 further
comprising:

a selector, having respective input terminals coupled to the first and
second packet stream sources, and an output terminal coupled to the protocol
decoder, and responsive to a select signal for coupling one of the first and
25 second packet stream sources to the protocol decoder; and wherein:

the third control program is responsive to the select signal to switch to the
first control program when the first packet stream source is coupled to the
protocol decoder and to switch to the second control program when the second
30 packet stream source is coupled to the protocol decoder.

11. The adaptive transport protocol decoder of claim 1 further comprising a payload processor coupled to the protocol decoder for processing the respective payloads extracted from the received packets.

5 12. The adaptive transport protocol decoder of claim 1 wherein:

each packet in the first and second packet streams further comprises a header containing information related to the payload;

10 the protocol decoder comprises a register for storing information from a header of a received packet; and

the protocol decoder accesses the register to obtain the information.

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

ABSTRACT OF THE DISCLOSURE

An adaptive transport protocol decoder includes a source of a stream of
5 packets, each including a payload, and having a first transport protocol, and a
source of a stream of packets, each including a payload, and having a second
transport protocol. A protocol decoder, coupled to the first and second packet
stream sources, extracts the respective payloads from the packets from a
selected one of the first and second packet stream sources.

02
03
04
05
06
07
08
09
10
11
12
13
14
15
16

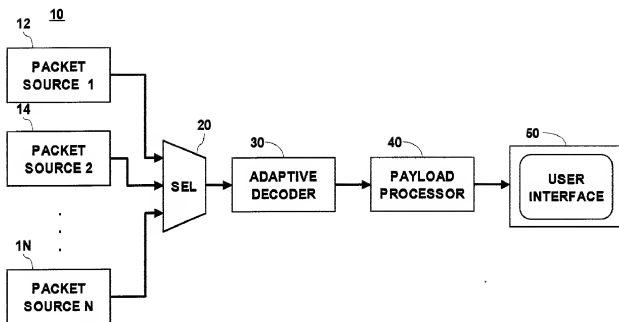


Fig. 1

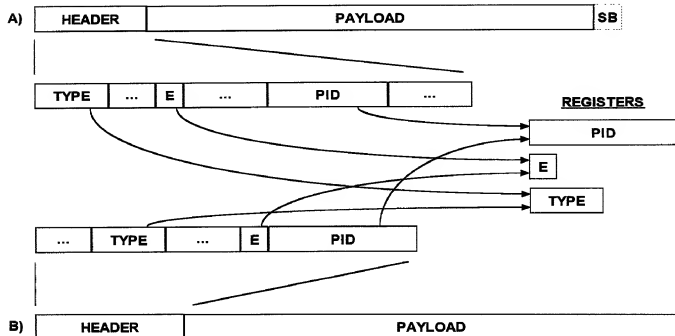


Fig. 2

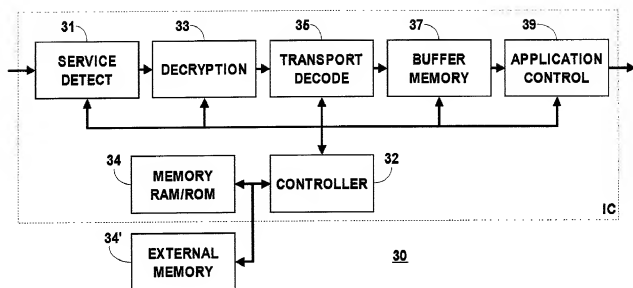


Fig. 3

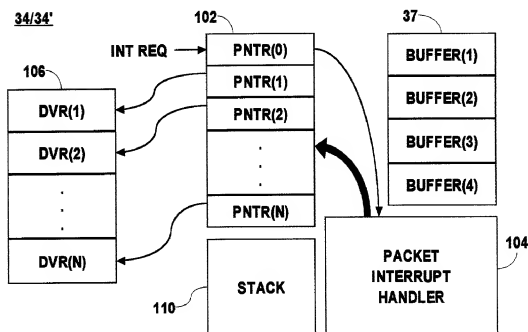


Fig. 4

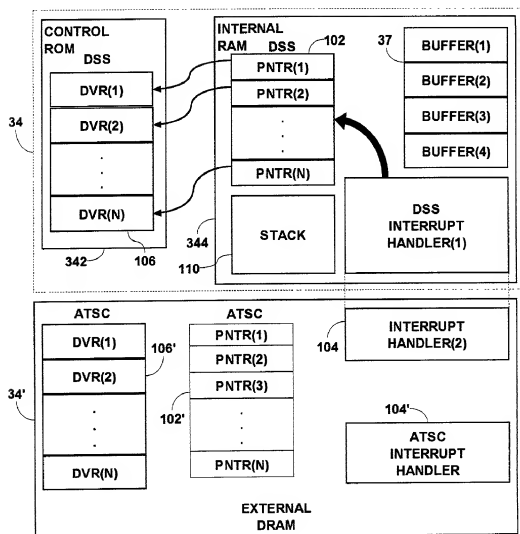


Fig. 5

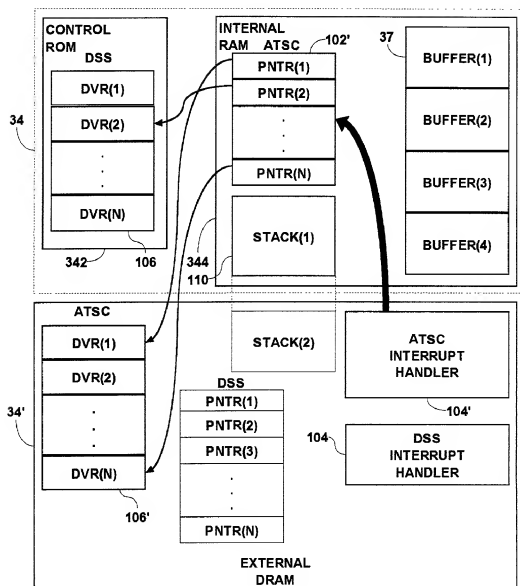
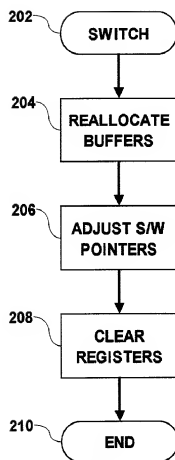


Fig. 6

**Fig. 7**

DECLARATION AND POWERS OF ATTORNEY

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

AN ADAPTIVE TRANSPORT PROTOCOL DECODER

the specification of which was filed on Herewith as Application Serial No. _____ and was amended on _____, or, if not identified here by filing date and serial number, is attached hereto.

I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with 37 CFR 1.56.

I hereby claim foreign priority benefits under 35 USC 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate by me or my representatives or assigns for this invention having a filing date before that of the application on which priority is claimed.

Application No. _____ in _____ on _____ priority claimed ☐ Yes ☐ No

Application No. _____ in _____ on _____ priority claimed ☐ Yes ☐ No

Application No. _____ in _____ on _____ priority claimed ☐ Yes ☐ No

I hereby claim the benefit under 35 USC 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of 35 USC 112, I acknowledge the duty to disclose material information as defined in 37 CFR 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

Serial No. _____ Filed _____ ☐ patented ☐ pending ☐ abandoned

Serial No. _____ Filed _____ ☐ patented ☐ pending ☐ abandoned

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 USC 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

I hereby appoint, individually and collectively, the following as my/our attorney or agent with full power of substitution and revocation, to prosecute this application and to transact all business in the U.S. Patent and Trademark Office connected therewith:

<u>Joseph S. Tripoli</u>	Registration No. <u>26,040</u>	and
<u>Eric P. Herrmann</u>	Registration No. <u>29,169</u>	and
<u>Ronald H. Kurdyla</u>	Registration No. <u>26,932</u>	and
	Registration No. _____	

PLEASE ADDRESS ALL

COMMUNICATIONS TO: JOSEPH S. TRIPOLI

PATENT OPERATIONS
THOMSON multimedia Licensing Inc.
PO Box 5312
PRINCETON, NEW JERSEY 08543-5312

Sole or Joint Inventor (1)	<u>Kevin Lloyd Grimes</u>	X <u>Kevin Lloyd Grimes</u>
	(Type or Print)	(Signature in Full. No Initials)
Citizenship	<u>U.S.A.</u>	Date X <u>10-20-99</u>
Post Office Address	<u>12427 Haydon Court, Fishers, Indiana 46038</u>	
Residence	<u>Fishers (Hamilton County), Indiana 46038</u>	
Sole or Joint Inventor (2)	<u>Kevin Elliot Bridgewater</u>	X <u>Kevin Elliot Bridgewater</u>
	(Type or Print)	(Signature in Full. No Initials)
Citizenship	<u>U.S.A.</u>	Date X <u>10-20-99</u>
Post Office Address	<u>290 South Muessing Road, Indianapolis, Indiana 46229</u>	
Residence	<u>Indianapolis (Marion County), Indiana 46229</u>	
Sole or Joint Inventor (3)	<u>Gregory George Tamer</u>	X <u>Gregory George Tamer</u>
	(Type or Print)	(Signature in Full. No Initials)
Citizenship	<u>U.S.A.</u>	Date X <u>10-20-99</u>
Post Office Address	<u>8447 Castle Ridge Lane, Indianapolis, Indiana 46256</u>	
Residence	<u>Indianapolis (Marion County), Indiana 46256</u>	

Sole or Joint Inventor (4)	<u>Thomas Edward Horlander</u> (Type or Print)	X <u>Thomas Edward Horlander</u> (Signature in Full. No initials.)
Citizenship	<u>U.S.A.</u>	Date X <u>10/20/99</u>
Post Office Address	<u>6234 Haverford Avenue, Indianapolis, Indiana 46220</u>	
Residence	<u>Indianapolis (Marion County), Indiana 46220</u>	
Sole or Joint Inventor (5)	<u>Todd Goosman</u> (Type or Print)	X <u>Todd Goosman</u> (Signature in Full. No initials.)
Citizenship	<u>U.S.A.</u>	Date X <u>10/21/99</u>
Post Office Address	<u>8217 Red Sail Court, Indianapolis, Indiana 46236</u>	
Residence	<u>Indianapolis (Marion County), Indiana 46236</u>	
Sole or Joint Inventor (6)	<u>Terry Wayne Lockridge</u> (Type or Print)	X <u>Terry Wayne Lockridge</u> (Signature in Full. No initials.)
Citizenship	<u>U.S.A.</u>	Date X <u>10/20/99</u>
Post Office Address	<u>10350 Ruckle, Indianapolis, Indiana 46280</u>	
Residence	<u>Indianapolis (Hamilton County), Indiana 46280</u>	